

JASPERREPORTS - REPORT PARAMETERS

http://www.tutorialspoint.com/jasper_reports/jasper_report_parameters.htm

Copyright © tutorialspoint.com

The main input for filling a report are: report template, parameters and datasources. This chapter will cover the parameters and in the next chapter we will cover the datasources.

Parameters are the object references that are passed during report-filling operations to the report engine. Parameters are useful for passing useful data to report engine, the data which cannot be passed through the datasource. Data like author name, title of the report etc, can be passed through parameters. A Jasper report template or JRXML template can have zero or more parameter elements.

Parameter Declaration

Parameter declaration is simple as follows:

```
<parameter name="exampleParameter" />
```

The name attribute

The *name* attribute of the <parameter> element is mandatory. It references the parameter in report expressions by name. Parameter name should be a single word. It should not contain any special characters like dot or comma.

The class attribute

The *class* attribute is also mandatory and it specifies the class name for the parameter values. Its default value is *java.lang.String*. This can be changed to any class available at runtime. Irrespective of the type of a report parameter, the engine takes care of casting in the report expressions in which the `$P{ }` token is used, hence making manual casts unnecessary.

The report parameter values are always packed in a *java.util.Map* object, which has the parameter name as its key. Report parameters can be used in the query string of the report, so as to further customize the data set retrieved from the database. These act like dynamic filters in the query that supplies data for the report.

Built-in Parameters

Following are the pre-defined report parameters, ready to use in expressions:

Parameter Name	Description
REPORT_PARAMETERS_MAP	Contains a map with all user defined and built-in parameters
REPORT_CONNECTION	This points to user supplied <i>java.sql.Connection</i> used for JDBC datasources
REPORT_DATA_SOURCE	This is a user supplied instance of <i>JRDataSource</i> representing either one of the built-in data source types or a user-defined one
REPORT_MAX_COUNT	This is a <i>java.lang.Integer</i> value, allowing the users to limit the records from datasource.
REPORT_SCRIPTLET	This points to <i>net.sf.jasperreports.engine.JRAbstractScriptlet</i> and contains an instance of the report scriptlet provided by the user

REPORT_LOCALE	This a <i>java.util.Locale</i> instance, containing the resource bundle desired locale
REPORT_RESOURCE_BUNDLE	This points to <i>java.util.ResourceBundle</i> object and contains localized messages
REPORT_TIME_ZONE	This a <i>java.util.TimeZone</i> instance, used for date formatting
REPORT_VIRTUALIZER	This is an instance of <i>net.sf.jasperreports.engine.JRVirtualizer</i> object, and used for page virtualization (optimize memory consumption)
REPORT_CLASS_LOADER	This is a <i>java.lang.ClassLoader</i> instance to be used during the report filling process to load resources such as images, fonts and subreport templates
IS_IGNORE_PAGINATION	If set to <i>java.lang.Boolean.TRUE</i> the report will be generated on one long page and page break will not occur

Example

Let us pass *ReportTitle* and *Author* to the report (generated by *JasperReportFill.java*). Revised file **C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\JasperReportFill.java** is as follows:

```
package com.tutorialspoint;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.Map;

import net.sf.jasperreports.engine.JRException;
import net.sf.jasperreports.engine.JasperFillManager;
import net.sf.jasperreports.engine.data.JRBeanCollectionDataSource;

public class JasperReportFill {
    @SuppressWarnings("unchecked")
    public static void main(String[] args) {
        String sourceFileName =
            "C://tools/jasperreports-5.0.1/test/jasper_report_template.jasper";

        DataBeanList DataBeanList = new DataBeanList();
        ArrayList<DataBean> dataList = DataBeanList.getDataBeanList();

        JRBeanCollectionDataSource beanColDataSource =
            new JRBeanCollectionDataSource(dataList);

        Map parameters = new HashMap();
        /**
         * Passing ReportTitle and Author as parameters
         */
        parameters.put("ReportTitle", "List of Contacts");
        parameters.put("Author", "Prepared By Manisha");

        try {
            JasperFillManager.fillReportToFile(
                sourceFileName, parameters, beanColDataSource);
        } catch (JRException e) {
            e.printStackTrace();
        }
    }
}
```

The contents of the POJO file **C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\DataBean.java** are as below:

```
package com.tutorialspoint;

public class DataBean {
    private String name;
```

```

private String country;

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public String getCountry() {
    return country;
}

public void setCountry(String country) {
    this.country = country;
}
}

```

The contents of the file **C:\tools\jasperreports-5.0.1\test\src\com\tutorialspoint\DataBeanList.java** are as below:

```

package com.tutorialspoint;

import java.util.ArrayList;

public class DataBeanList {
    public ArrayList<DataBean> getDataBeanList() {
        ArrayList<DataBean> dataBeanList = new ArrayList<DataBean>();

        dataBeanList.add(produce("Manisha", "India"));
        dataBeanList.add(produce("Dennis Ritchie", "USA"));
        dataBeanList.add(produce("V.Anand", "India"));
        dataBeanList.add(produce("Shrinath", "California"));

        return dataBeanList;
    }

    /**
     * This method returns a DataBean object,
     * with name and country set in it.
     */
    private DataBean produce(String name, String country) {
        DataBean dataBean = new DataBean();
        dataBean.setName(name);
        dataBean.setCountry(country);
        return dataBean;
    }
}

```

Let's add parameters the **<ReportTitle>** and **<Author>** to our existing report template (Chapter [Report designs](#)). The Report Title and Author will be displayed at the beginning of the report. The revised report template (jasper_report_template.jrxml) is as follows. Save it to C:\tools\jasperreports-5.0.1\test directory:

```

<?xml version="1.0"?>
<!DOCTYPE jasperReport PUBLIC
"//JasperReports//DTD Report Design//EN"
"http://jasperreports.sourceforge.net/dtds/jasperreport.dtd">

<jasperReport xmlns="http://jasperreports.sourceforge.net/jasperreports"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://jasperreports.sourceforge.net/jasperreports
http://jasperreports.sourceforge.net/xsd/jasperreport.xsd"
name="jasper_report_template" pageWidth="595"
pageHeight="842" columnWidth="515"
leftMargin="40" rightMargin="40" topMargin="50" bottomMargin="50">
<parameter name="ReportTitle" />
<parameter name="Author" />

<queryString>

```

```

<![CDATA[]]>
</queryString>

<field name="country" >
  <fieldDescription><![CDATA[country]]></fieldDescription>
</field>

<field name="name" >
  <fieldDescription><![CDATA[name]]></fieldDescription>
</field>

<title>
  <band height="70">
    <line>
      <reportElement x="0" y="0" width="515"
        height="1"/>
    </line>
    <textField isBlankWhenNull="true" bookmarkLevel="1">
      <reportElement x="0" y="10" width="515"
        height="30"/>
      <textElement textAlignment="Center">
        <font size="22"/>
      </textElement>
      <textFieldExpression >
        <![CDATA[${P{ReportTitle}}]>
      </textFieldExpression>
      <anchorNameExpression><![CDATA["Title"]]>
      </anchorNameExpression>
    </textField>
    <textField isBlankWhenNull="true">
      <reportElement x="0" y="40" width="515" height="20"/>
      <textElement textAlignment="Center">
        <font size="10"/>
      </textElement>
      <textFieldExpression >
        <![CDATA[${P{Author}}]>
      </textFieldExpression>
    </textField>
  </band>
</title>

<columnHeader>
  <band height="23">
    <staticText>
      <reportElement mode="Opaque" x="0" y="3"
        width="535" height="15"
        backcolor="#70A9A9" />
      <box>
        <bottomPen lineWidth="1.0"
          lineColor="#CCCCCC" />
      </box>
      <textElement />
      <text><![CDATA[]]>
      </text>
    </staticText>
    <staticText>
      <reportElement x="414" y="3" width="121"
        height="15" />
      <textElement textAlignment="Center"
        verticalAlignment="Middle">
        <font isBold="true" />
      </textElement>
      <text><![CDATA[Country]]></text>
    </staticText>
    <staticText>
      <reportElement x="0" y="3" width="136"
        height="15" />
      <textElement textAlignment="Center"
        verticalAlignment="Middle">
        <font isBold="true" />
      </textElement>
      <text><![CDATA[Name]]></text>
    </staticText>
  </band>
</columnHeader>

```

```

        </band>
    </columnHeader>

    <detail>
        <band height="16">
            <staticText>
                <reportElement mode="Opaque" x="0" y="0"
                width="535" height="14"
                backcolor="#E5ECF9" />
                <box>
                    <bottomPen lineWidth="0.25"
                    lineColor="#CCCCCC" />
                </box>
                <textElement />
                <text><![CDATA[]]>
                </text>
            </staticText>
            <textField>
                <reportElement x="414" y="0" width="121"
                height="15" />
                <textElement textAlignment="Center"
                verticalAlignment="Middle">
                    <font size="9" />
                </textElement>
                <textFieldExpression >
                <![CDATA[${F{country}}]>
                </textFieldExpression>
            </textField>
            <textField>
                <reportElement x="0" y="0" width="136"
                height="15" />
                <textElement textAlignment="Center"
                verticalAlignment="Middle" />
                <textFieldExpression >
                <![CDATA[${F{name}}]>
                </textFieldExpression>
            </textField>
        </band>
    </detail>
</jasperReport>

```

Report Generation

We will compile and execute the above file using our regular ANT build process. The contents of the file build.xml (saved under directory C:\tools\jasperreports-5.0.1\test) are as below.

The import file - baseBuild.xml is picked from chapter [Environment Setup](#) and should be placed in the same directory as the build.xml.

```

<?xml version="1.0" encoding="UTF-8"?>
<project name="JasperReportTest" default="viewFillReport" basedir=".">
    <import file="baseBuild.xml" />
    <target name="viewFillReport"
        depends="compile,compilereportdesing,run"
        description="Launches the report viewer to preview
        the report stored in the .JRprint file.">
        <java classname="net.sf.jasperreports.view.JasperViewer"
        fork="true">
            <arg value="-F${file.name}.JRprint" />
            <classpath ref />
        </java>
    </target>
    <target name="compilereportdesing"
        description="Compiles the JXML file and
        produces the .jasper file.">
        <taskdef name="jrc"

```

```

        classname="net.sf.jasperreports.ant.JRAntCompileTask">
        <classpath ref />
    </taskdef>
    <jrc destdir=".">
        <src>
        <fileset dir=".">
            <include name="*.jrxml" />
        </fileset>
        </src>
        <classpath ref />
    </jrc>
</target>
</project>

```

Next, let's open command line window and go to the directory where build.xml is placed. Finally execute the command **ant -Dmain-class=com.tutorialspoint.JasperReportFill** (viewFullReport is the default target) as follows:

```

C:\tools\jasperreports-5.0.1\test>ant -Dmain-class=com.tutorialspoint.JasperReportFill
Buildfile: C:\tools\jasperreports-5.0.1\test\build.xml

clean-sample:
[delete] Deleting directory C:\tools\jasperreports-5.0.1\test\classes
[delete] Deleting: C:\tools\jasperreports-5.0.1\test\jasper_report_template.jasper
[delete] Deleting: C:\tools\jasperreports-5.0.1\test\jasper_report_template.jrprint

compile:
[mkdir] Created dir: C:\tools\jasperreports-5.0.1\test\classes
[javac] C:\tools\jasperreports-5.0.1\test\baseBuild.xml:28: warning:
'incldeantruntime' was not set, defaulting to build.sysclasspath=last;
set to false for repeatable builds
[javac] Compiling 7 source files to C:\tools\jasperreports-5.0.1\test\classes

compilereportdesing:
[jrc] Compiling 1 report design files.
[jrc] log4j:WARN No appenders could be found for logger
(net.sf.jasperreports.engine.xml.JRXmlDigesterFactory).
[jrc] log4j:WARN Please initialize the log4j system properly.
[jrc] log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig
for more info.
[jrc] File : C:\tools\jasperreports-5.0.1\test\jasper_report_template.jrxml ... OK.

run:
[echo] Runnin class : com.tutorialspoint.JasperReportFill
[java] log4j:WARN No appenders could be found for logger
(net.sf.jasperreports.extensions.ExtensionsEnvironment).
[java] log4j:WARN Please initialize the log4j system properly.

viewFillReport:
[java] log4j:WARN No appenders could be found for logger
(net.sf.jasperreports.extensions.ExtensionsEnvironment).
[java] log4j:WARN Please initialize the log4j system properly.

BUILD SUCCESSFUL
Total time: 18 seconds

```

As a result of above compilation, a JasperViewer window opens up as in the screen below:

List of Contacts
Prepared By Manisha

Name	Country
Manisha	India
Dennis Ritchie	USA
V. Anand	India
Shrinath	California

Page 1 of 1

Here we see that, the ReportTitle "List Of Contacts" and Author "Prepared By Manisha" are displayed at the beginning of the report.